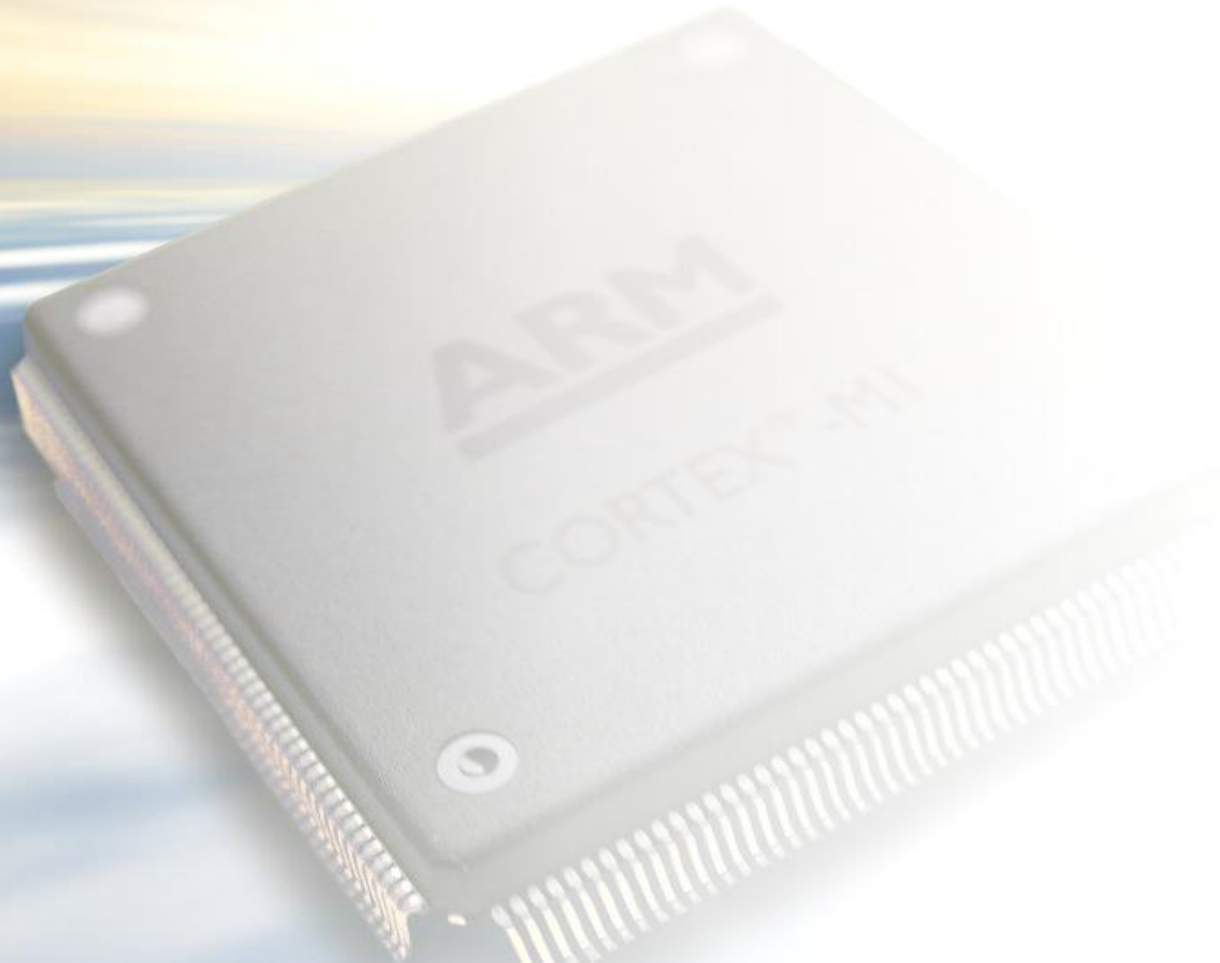


Functional Safety – How ARM[®] solutions benefit your design



The Architecture for the Digital World[®]

ARM

Functional Safety - required in many markets

Automotive



Industrial



Medical



Railway



ARM's package to support your design

- **ARM® Cortex®-R5 processor** - safety document package
- **ARM® Compiler** – safety package for software development
 - Qualification Kit
 - Extended Maintenance
 - Functional Safety Certified

ARM® Cortex®-R5 Processor - Safety Document Package

Cortex-R5 Safety Package

- **Safety Manual**
- Failure Modes and Effects Analysis
- Development Interface Report



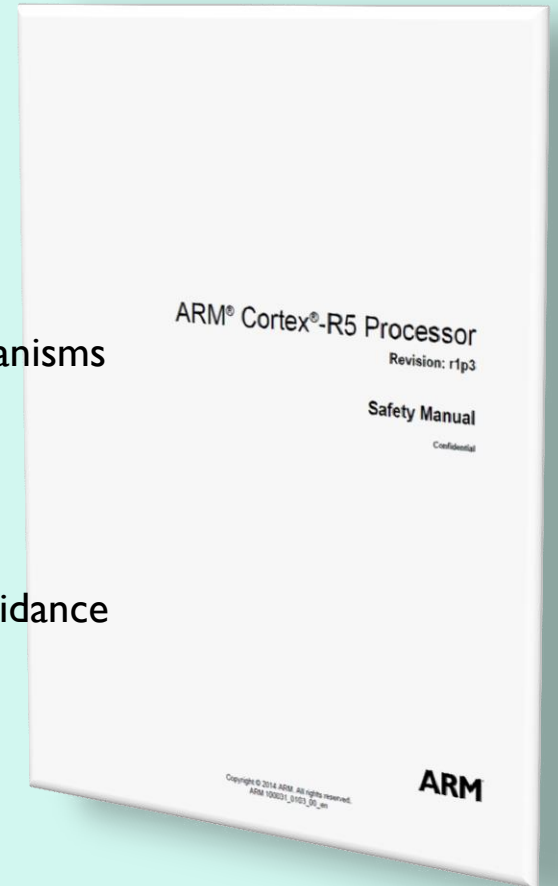
Safety Package

Cortex-R5 **Safety Manual:**

- Introduction
- Cortex-R5 safety lifecycle
- Cortex-R5 safety architecture
- Cortex-R5 configuration options
- Cortex-R5 fault detection and control mechanisms
- Cortex-R5 assumptions of use
- Cortex-R5 safety analysis results

- Appendix - ECC tables
- Appendix - Measures for systematic fault avoidance
- Appendix - Lock-step initialization sequence

- Total contents approx. 150 pages



Safety documentation on other selected Cortex processors to follow

ARM® Compiler - Safety Package

- Compiler Safety Package for software development in safety markets
 - Industrial control, automotive, medical, transportation, military and others

Qualification Kit

- Development process docs
- Safety manual
- Defect report
- Test report



Extended Maintenance

- Five year commitment
- Technical support
- Critical defect fixes



Functional Safety Certified

- TÜV SÜD certification
- ISO 26262 (ASILD)
- IEC 61508 (SIL3)



- The Safety Package comes with DS-5 Ultimate and Keil MDK-Professional
- For more info please visit
 - <http://ds.arm.com/solutions/safety-critical/>
 - <http://www2.keil.com/safety/>

ARM® Compiler - Qualification Kit - toolchain justification

ARM Compiler Qualification Kit

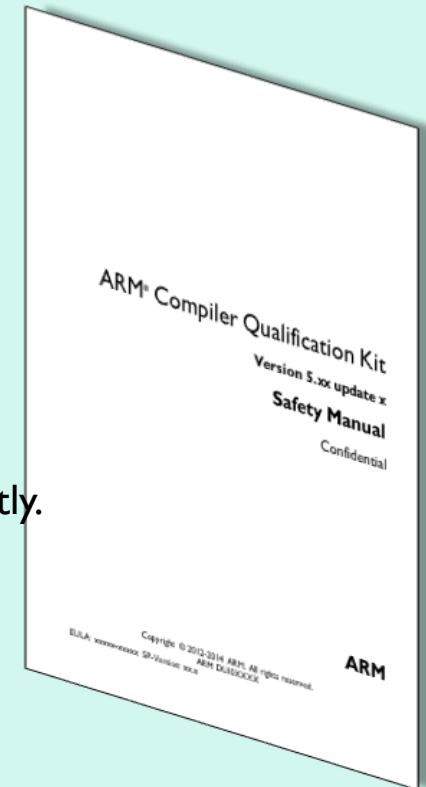
- **Safety Manual**
- Development Process
- Test Report
- Defect Report
- Failure Modes and Effects Analysis
- Release History



Qualification Kit

ARM Compiler **Safety Manual**:

- Description of the
 - high level safety-related faults that the ARM Compiler can generate.
 - communication between the tools, which shows how faults might propagate.
- Mitigation strategies that can be followed
 - to prevent or work-around faults.
 - for identification of specific behaviours of the ARM Compiler that are unsafe if used incorrectly.
- Total contents approx. 120 pages

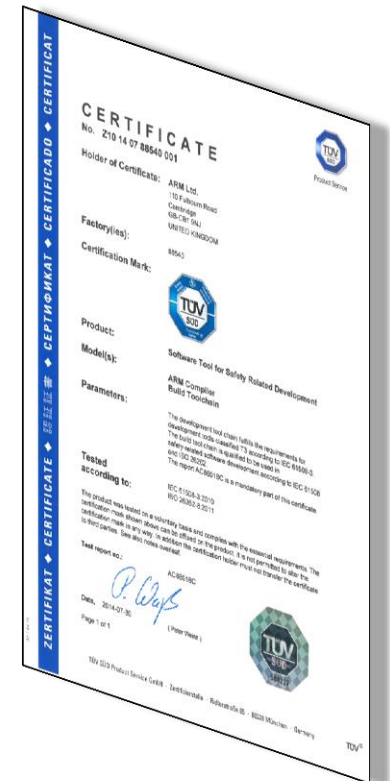


ARM® Compiler - Extended Maintenance (EM)

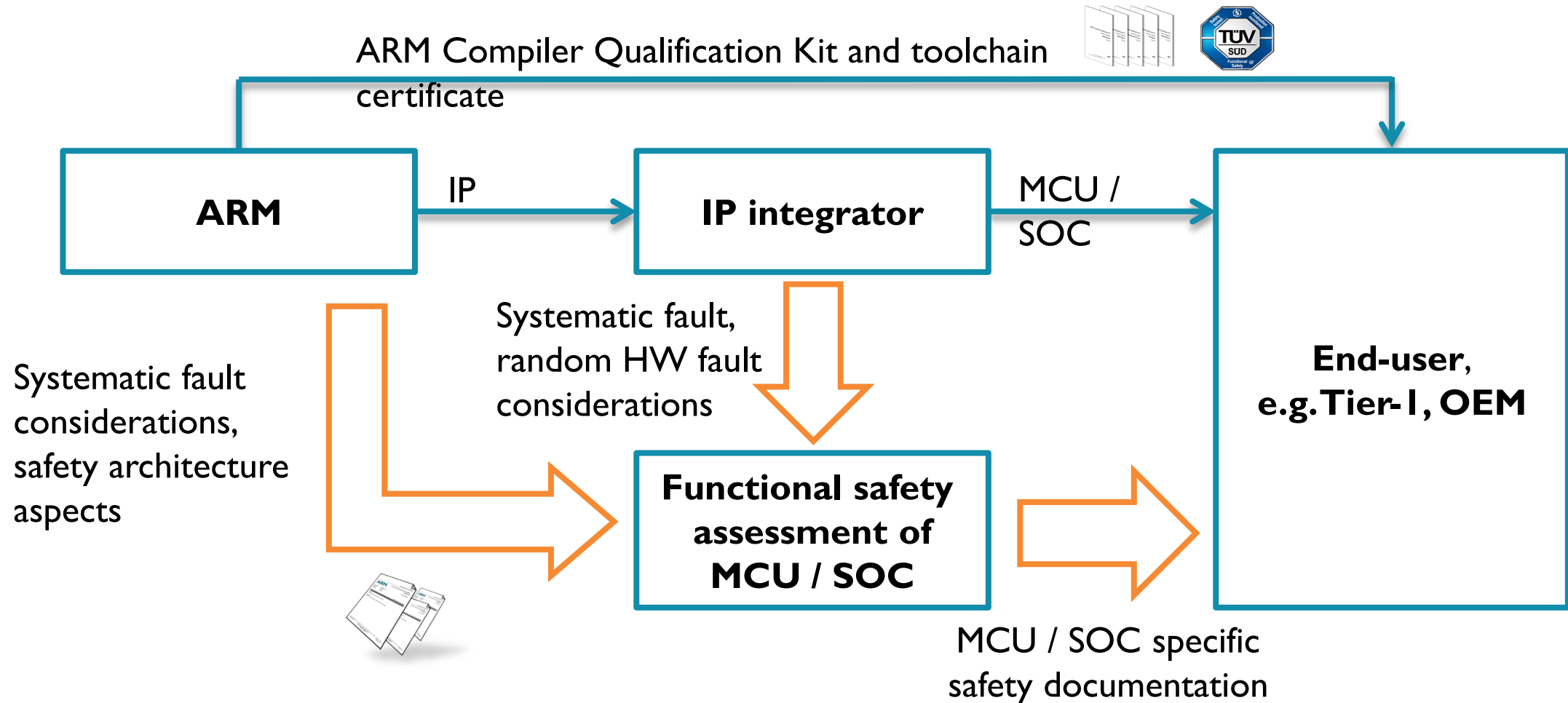
- ARM Compiler 5.04 is a stable branch with regularly scheduled maintenance releases.
 - New critical defects are back-ported to EM branch.
- ARM Compiler 5.04 is the first EM branch, followed by ARM Compiler 5.06 in 2015.
 - ARM Compiler 5.04 will be maintained through 2015.
 - ARM Compiler 5.06 will be maintained through 2020,
 - longer term contracts may be available by special arrangement with ARM.
- ARM Compiler Qualification Kit aligns with EM branches.
 - For every Extended Maintenance release, there will be a QK update.
 - made available to registered users at connect.arm.com

ARM® Compiler - Functional Safety Certified

- ARM Compiler 5.04u2 and future ARM Compiler 5.04 releases will be TÜV SÜD certified
 - as meeting the toolchain requirements of ISO 26262 (through ASIL D) and IEC 61508 (through SIL 3),
 - as being qualified for safety-related software development up to SIL3 (IEC 61508) or ASILD (ISO 26262) without further qualification activities,
 - and will be licensed as part of MDK-Professional and DS-5 Ultimate package.



Functional Safety – How ARM® solutions benefit your design



MDK: Features to Support Verification and Certification

Debug & Trace for Cortex-M

- Code Coverage
- Exception Trace
- Event Viewer
- Code Instrumentation

The screenshot displays three windows from the MDK software:

- Trace Data:** A table showing execution details. The first row shows an LDR instruction at address 0x080003FA. The second row shows a BX instruction at address 0x080003FC. The function being traced is HAL_ADC_PollForConver...
- Disassembly:** A window showing assembly code for the function ADC_StartConversion(). It includes instructions like BL.W, MOV, LDR, EOR, CBZ, and MOV, along with their corresponding addresses and comments.
- Event Viewer:** A timeline view showing the execution of threads. The main thread (main (2)) is shown with several execution blocks. Other threads like osTimerThread (1), SVCall (11), and various DMA and I2C streams are also visible.

Thank You



The trademarks featured in this presentation are registered and/or unregistered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Any other marks featured may be trademarks of their respective owners